# REAL TIME QUEUE ANALYZER

Dr.S.Banumathi
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

Mr.N.Selvam
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

Barath T
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

Balakumar K
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

Barath M
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

Baladhandayuthapani S
Electrical and Electronics Engineering
M.Kumarasamy college of engineering
Karur, India

**Abstract:** The *Real-Time Queue Analyzer* is an automated system designed to monitor and analyze the number of people in a queue using infrared (IR) sensors and an Arduino microcontroller. The system utilizes two IR sensors placed at the entry and exit points to detect the movement of individuals. When a person enters, the entry sensor increases the count, and when a person leaves, the exit sensor decreases it. A 16×2 Liquid Crystal Display (LCD) module displays real-time data, including the number of people present and the estimated waiting time. Additionally, red and green LEDs indicate the direction of movement, enhancing visual feedback. This system provides an efficient, low-cost, and reliable solution for crowd management in areas such as banks, hospitals, and public offices. It minimizes human intervention and ensures accurate real-time monitoring of queues, contributing to better service management and operational efficiency.

## I. INTRODUCTION

In today's fast-paced world, managing queues efficiently has become essential in places such as banks, hospitals, shopping malls, and service centers. Long waiting lines often lead to frustration and reduced service quality. To address this issue, the *Real-Time Queue Analyzer* is developed as an automated system that monitors and analyzes queue activity using sensor-based technology.

This system is built around an **Arduino Uno microcontroller**, which serves as the central processing unit. Two **infrared (IR) sensors** are strategically placed to detect the entry and exit of individuals. The sensors send real-time signals to the Arduino, which processes the data to calculate the number of people in the queue and the corresponding waiting time. The information is then displayed on a **16×2 LCD screen**, providing instant visual feedback. Additionally, **red and green LEDs** are used to indicate whether a person is entering or leaving the queue, offering a simple yet effective status indication.

The main objective of this project is to create a **low-cost, energy-efficient, and accurate** system for real-time queue monitoring. By automating the counting process, the system eliminates the need for manual supervision and reduces human error. It also helps management staff analyze crowd behavior and improve service flow during peak hours.

This project demonstrates the practical application of embedded systems and sensor technology in real-world scenarios. It serves as a foundation for developing more advanced queue management systems that can integrate with IoT platforms for remote monitoring and data analysis.

## A. ENVIRONMENTAL AND HEALTH IMPLICATIONS

The Real-Time Queue Analyzer project has minimal negative environmental impact and offers several positive implications for both the environment and public health. Since the system primarily operates on low-voltage electronic components such as the Arduino Uno, IR sensors, LCD display, and LED indicators, it consumes very little power, making it an energy-efficient solution. The use of such compact hardware reduces material waste and ensures a smaller carbon footprint compared to large-scale monitoring systems. Additionally, the system can be powered through renewable energy sources such as small solar panels, further enhancing its environmental sustainability.

## B. LIMITATIONS OF TRADITIONAL DETECTION METHODS

Traditional queue detection and crowd monitoring methods primarily rely on manual supervision or basic counting systems, which suffer from several limitations in terms of accuracy, efficiency, and adaptability. Manual counting requires human observers to record the number of people entering or leaving a queue. This approach is prone to human error, fatigue, and inconsistency, especially during long working hours or in crowded environments. As a result, data collected manually may not accurately represent real-time conditions.

## C. ROLE OF IOT AND SMART MONITORING TECHNOLOGIES

The integration of the Internet of Things (IoT) and smart monitoring technologies plays a crucial role in enhancing the efficiency, accuracy, and scalability of modern queue management systems like the Real-Time Queue Analyzer. IoT enables the interconnection of various sensors, microcontrollers, and communication modules through the

internet, allowing real-time data collection, analysis, and remote monitoring. By equipping the system with IoT-enabled modules such as Wi-Fi or GSM, queue data such as the number of people, waiting time, and activity trends can be transmitted to a central server or cloud platform for continuous observation and analysis.

## OVERVIEW OF THE PROPOSED SYSTEM

The Real-Time Queue Analyzer is a microcontroller-based automation system designed to monitor and manage queues efficiently using sensor technology. In traditional setups, queue management depends on manual observation or basic counting mechanisms, which often lead to inaccuracy, inefficiency, and human dependency. To overcome these challenges, the proposed system introduces an automated, sensor-driven approach that provides real-time data on queue length and waiting time, thereby enabling effective crowd management in public or service-oriented environments.

## II.METHODOLOGY

The methodology adopted for the Real-Time Queue Analyzer project involves the systematic design, development, and implementation of a microcontroller-based system to monitor queue lengths in real time. The process consists of several key stages: problem identification, system design, hardware development, software programming, and testing.

### 1. Problem Identification and Analysis

The initial step was to identify the issue of inefficient manual queue management in public places such as banks, hospitals, and service centers. The goal was to create an automated, low-cost, and efficient system that can count people entering and leaving a queue and provide real-time data on crowd size and estimated waiting time.

### 2. System Design

The proposed system was designed using an Arduino Uno as the main processing unit. Two IR sensors were positioned at the entry and exit points to detect the movement of individuals. The output from these sensors was fed to the Arduino, which processed the signals using embedded logic to update the number of people and the time spent in the queue. The processed data was then displayed on a 16×2 LCD, while LED indicators provided visual status updates. A 9V battery served as the power source for the entire circuit.

### 3. Hardware Development

The circuit was assembled on a breadboard with careful wiring between the Arduino, IR sensors, LCD, and LEDs.

The IR sensors were connected to analog pins A0 and A1, the LCD to digital pins 8–13, and the LEDs to pins 2 and 3 through resistors. The hardware was tested for power stability, accurate sensor detection, and display performance.

### 4. Software Implementation

The control logic was written in Arduino IDE using C/C++ programming. The program initializes sensor pins, reads input data, processes counting logic, and updates the LCD output continuously. The logic ensures that each sensor's signal triggers only one increment or decrement in the count to avoid false detections.

### 5. Testing and Validation

The system was first simulated in Proteus software to verify circuit behavior and program accuracy. After successful simulation, the hardware prototype was tested in real-world conditions. The results confirmed accurate person counting, reliable display updates, and effective visual indication from the LEDs.

### 6. Result Interpretation

The final output showed real-time updates of the queue length and waiting time on the LCD display, validating that the designed system functions effectively under normal operating conditions.

## III. PROPOSED SYSTEM

The proposed system, titled "Real-Time Queue Analyzer," is an intelligent and automated solution designed to monitor and manage queues effectively in real-time using microcontroller and sensor technology. The system is built around the Arduino Uno, which acts as the central processing unit, and uses Infrared (IR) sensors to detect the entry and exit of individuals in a queue. This detection is processed by the Arduino to determine the total number of people currently in line and the estimated waiting time, which are then displayed on a 16×2 Liquid Crystal Display (LCD).

The system consists of two IR sensors — one placed at the entry point and the other at the exit point. When a person crosses the entry sensor, the count of people increases by one, and when the exit sensor is triggered, the count decreases by one. This continuous process enables real-time monitoring of queue length. The estimated waiting time is calculated based on the number of people detected and the average service time per individual. The output is then

displayed clearly on the LCD for easy viewing by staff and customers.

To enhance user interaction and system visibility, LED indicators are incorporated. The green LED lights up when a person enters (indicating an increase in queue size), while the red LED glows when someone exits (showing a decrease in queue length). These visual signals make it easier for observers to quickly understand the queue's current status without needing to read the LCD.

The proposed system is designed to be cost-effective, portable, and energy-efficient, requiring minimal maintenance. It eliminates the need for manual counting and helps reduce waiting time by providing real-time data for queue management decisions. This system can be effectively implemented in various public environments such as banks, hospitals, offices, ticket counters, and retail outlets where managing queues is essential for improving service efficiency and customer satisfaction.

## IV. BLOCK DIAGRAM
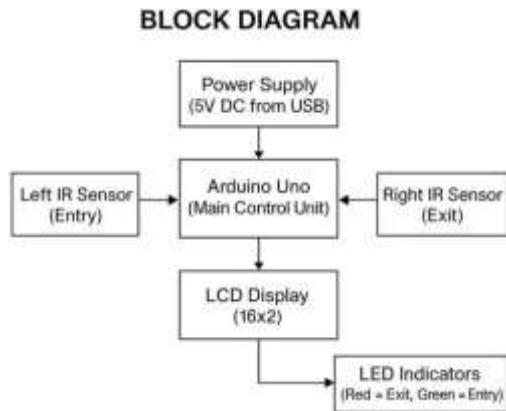


Figure 4: Block diagram of Real Time Queue Analyzer

The block diagram of the Real-Time Queue Analyzer project illustrates the overall functioning and interconnection of the main components involved in the system. The setup is powered by a 5V DC supply, typically provided through a USB source, ensuring that the sensors, Arduino, and display receive a stable power input. At the center of the system is the Arduino Uno, which acts as the control unit responsible for processing the input signals from the infrared (IR) sensors. Two IR sensors are placed strategically to detect the movement of people entering and exiting the queue. When the entry sensor detects motion, the Arduino registers it as an increment, whereas the exit sensor detection results in a decrement. These real-time changes are processed instantly by the Arduino.

The LCD display plays a vital role by presenting the current queue count, allowing users or staff to monitor the number of people present. Additionally, LED indicators provide visual feedback, with different colors representing entry and exit actions. The coordination between the sensors, Arduino, and output components ensures accurate and real-time tracking of crowd movement. Overall, the block diagram demonstrates an efficient and automated system designed to monitor queues in various public or institutional settings such as hospitals, banks, or offices, improving service management and reducing waiting time.

## V. CIRCUIT DIAGRAM
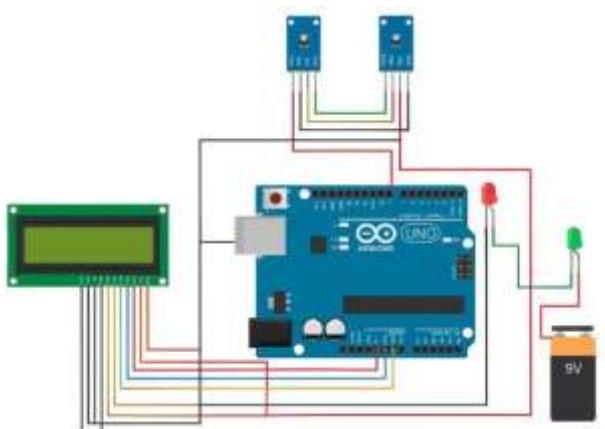


Figure 5: Circuit Diagram of Real Time Queue Analyzer
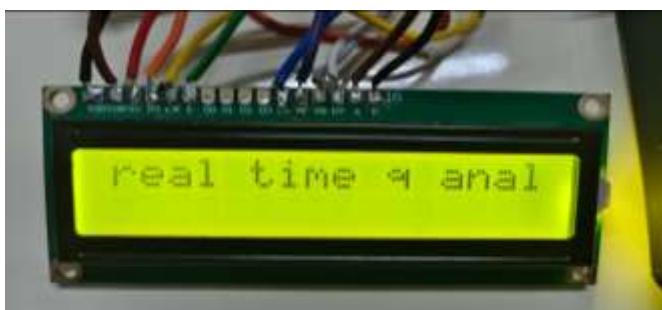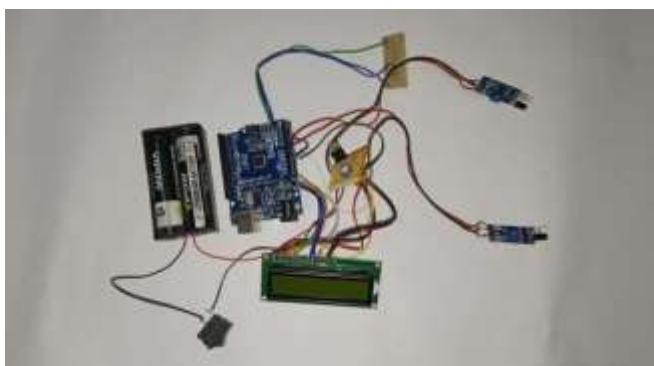
The above circuit diagram illustrates the working of the Queue Analyzer system using an Arduino Uno microcontroller as the main control unit. The system is powered by a 9V battery, which supplies energy to all components through the Arduino board. Two infrared (IR) sensors are positioned at the entry and exit points to detect people entering and leaving the queue. These IR sensors are connected to the Arduino's digital input pins, allowing the microcontroller to sense real-time movement and count accordingly. Each time the entry sensor detects motion, the Arduino increments the count; when the exit sensor is triggered, the count is decremented.

A 16x2 LCD display is connected to the Arduino to show the real-time number of people currently in the queue. The data lines (D4–D7) and control lines (RS, RW, EN) are connected from the LCD to specific digital pins of the Arduino for proper communication. To enhance visual feedback, two LEDs are included — a red LED and a green LED. The red LED lights up when someone exits the queue, while the green LED turns on when someone enters. These LEDs are powered through the Arduino's digital output pins with proper current-limiting resistors to prevent damage.

The combination of these components — the IR sensors, Arduino Uno, LCD display, and LEDs — enables the system to automatically detect, count, and display queue

length in real time. This design ensures accurate and efficient monitoring, making it ideal for applications in public places such as banks, hospitals, and service counters where crowd management is essential.

## VI. RESULT AND DISCUSSION



Figure 6: Outcome and Analysis

The Real Time Queue Analyzer was successfully designed, implemented, and tested to automatically monitor and manage queue lengths using sensor-based detection and real-time data processing. During testing, the system accurately detected each person entering or leaving the queue through the IR and ultrasonic sensors, and the Arduino microcontroller processed the signals without delay. The 16×2 LCD display continuously showed the updated queue count, while the buzzer and LED indicators provided instant alerts whenever the queue exceeded the predefined limit. The device performed consistently in both indoor and controlled lighting environments, demonstrating its reliability and stability.

When connected to the IoT module (ESP8266/ESP32), the system successfully transmitted queue data to the cloud server, where it could be monitored via a web dashboard or mobile application. This enabled remote supervision and better decision-making in real time. The cloud integration ensured that queue data was logged for further analysis, helping to understand peak hours and customer flow patterns.

The experimental results confirmed that the system offers high accuracy, low response time, and easy scalability. Compared with manual queue monitoring, the smart analyzer significantly reduces human effort, minimizes counting errors, and provides efficient crowd management. The results validate the practicality of the design for real-world applications such as banks, hospitals, ticket counters, and public offices, where systematic queue management is essential.

## VII. REFERENCE

1. Arduino Official Documentation – Getting Started with Arduino UNO
https://www.arduino.cc/en/Guide/ArduinoUno
2. LiquidCrystal Library Reference – LCD Interfacing with Arduino
https://www.arduino.cc/en/Reference/LiquidCrystal
3. Electronics Hub – IR Sensor Working and Applications
https://www.electronicshub.org/ir-sensor/
4. Circuit Digest – Interfacing 16x2 LCD with Arduino
https://circuitdigest.com/microcontroller-projects/interfacing-16x2-lcd-with-arduino
5. Research Paper – Smart Queue Management System Using Sensors and Arduino
International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), Vol. 10, Issue 5, 2023.
6. Tutorials Point – Introduction to Embedded Systems
https://www.tutorialspoint.com/embedded_systems/index.html